Remarks

The present response is to the Office Action mailed in the above referenced case on May 15, 2007, made Final. Claims 1-28 are standing for examination. Claims 1-28 are rejected under 35 U.S.C. 103(a) as being unpatentable over DaCosta et al. (US 6,826,553) hereinafter DaCosta in view of Weinberg et al. (US 6,360,332) hereinafter Weinberg.

In response applicant herein provides more detailed arguments which clearly show that all of the limitations as recited in applicant's claims are not taught or suggested in the art provided by the Examiner.

Regarding the 103 rejection of claims 1 and 12, the Examiner states; "in regard to substantially similar independent claims 1 and 12, DaCosta teaches an application for enabling automated notification of applied structural changes to electronic information pages on a network comprising: an interface for enabling users to build and modify network navigation and interaction templates using functional logic blocks for automatically navigating to and interacting with interactive electronic information pages on the network (column 2, lines 11-30; column 5, lines 30-67)(Figs. 1 & 7);"

Applicant argues that DaCosta teaches recording portions of interactions to create scripts. This teaching does not read on the functional logic blocks as taught and recited in applicant's invention as claimed.

Specifically, the portions of DaCosta relied upon by the Examiner recite; *"One embodiment of the present invention provides a system for collecting unstructured data from one or more web sites on the Internet and providing structured data, for example, to navigate to multiple web sites and extract data snippets. The system in accordance with one embodiment of the present invention enables the process of collecting such data to be* *automated so that one or more target data sources can be constantly monitored. In accordance with a preferred embodiment of the present invention, the data location and scraping tool of the present invention comprises a browser plug-in to facilitate data collection, for example, scripts are added to the browser such as Microsoft Internet Explorer. Thus, the browser effectively serves as the operating system, and the scripts embedded in the browser form an input layer that locates and extracts data and effectively serves as a BIOS for retrieval of unstructured data. The data can be simply displayed or imported and stored in a database, for example, or can be further processed, for example, using a spreadsheet application, and even imported directly to one or more applications." (col. 2, lines 11-30)*

Applicant cannot discern which element in the DaCosta description above the Examiner has interpreted as applicant's claimed functional logic blocks. With no explanation offered by the Examiner, applicant is clearly at a disadvantage. Applicant will point out that retrieving unstructured data and converting it into structured data in order for a Web developer to write a script cannot read on applicant's claimed functional logic blocks. Applicant argues that the scripts taught in DaCosta also fail to read on the functional logic blocks. Applicant also cannot interpret any element in Fig. 1 and 7 which read on providing a change-notification module for indicating a point in process where a navigation and interaction routine has failed and for creating a data file containing parameters associated with the failed routine.

Column 5, lines 30-67 of DaCosta, relied upon by the Examiner recites; *"Referring now to FIG. 1, one embodiment of the present invention provides a core layer on top of which automated web applications can be built. In other words, the present invention provides a tool, that can be used by developers or web users with multiple potential different interfaces, to access data available within of one or more web pages or other web-accessible documents in an automated fashion.*

Considered in more detail, a navigation Application Program Interface ("API")

10 enables a client application program running on a microprocessor-based device of a user to learn and store navigation paths to given web pages or other web-accessible documents, including dialogs and forms that need to be filled in to reach those locations or sites, for example. The navigation API 10 includes a recording module 12 and playback module 14. For example, if a web site requires a user to enter a login name and password to reach an orientation page and then asks for a set of preferences to go to specific web pages or other web- accessible documents of interest, it is an object of the present invention to enable a client application to record this path once, then play it back many times including the dialog interaction with the server. In a generic example, this could allow one to record "metabookmarks", i.e., bookmarks that record not only a destination Uniform Resource Locator ("URL"), but also the required steps to navigate thereto, and play those steps back. Additionally, as shown in FIG. 1, an extraction API 20 enables an application to robustly define data segments in a web page or other web-accessible document. Similar to the case of the navigation module 10, there are recording and playback modules 22 and 24 included in the extraction API 20. However, instead of recording playing across web pages and web space, the extraction module 20 records and plays across elements within a single page. Artificial intelligence ("AI") techniques can be utilized to enable pattern matching to ensure that the relevant information will still be retrieved even if the page is modified (within reasonable limits, of course)."

Applicant argues that the above portion of DaCosta also fails to read on a developer-interface module for enabling developers to build and modify network navigation and interaction templates using functional logic blocks, for automatically navigating to and interacting with interactive electronic information pages on the data packet network, as claimed. The above portion of DaCosta teaches a method of utilizing an API on a user's computer to learn and store navigation paths for sites using a recording module 12 and a playback module 14. Metabookmarks are also incorporated to store a URL and recorded steps to play back in order to navigate the Web site.

Applicant claims an ability to build and modify network navigation and interaction templates (navigation orders) using functional logic blocks, meaning software logic blocks including function, not recorded portions of navigation as in DaCosta. DaCosta fails to teach creating navigation scripts using logic blocks, as claimed. In applicant's invention a site logic portion of a navigation script may contain more than one identifiable interaction task. Therefore, site-logic blocks, which are modular parts of whole navigation orders contain all of the possible interaction instructions available at the associated site. In a given navigation order, the associated site-logic blocks are activated to enable only the specified interactions described in the request portion of the order specified by the requesting user. The modular concept used in constructing navigation orders enables site-logic portions as well as login portions and other functional portions of the order to be interchanged in an efficient manner. Applicant argues, as evidenced above, DaCosta fails to teach using functional logic blocks, as claimed.

The Examiner states DaCosta teaches "a change notification module for indicating a point in process where a navigation and interaction routine has failed and for creating a data file containing parameters associated with the failed routine (column 6, lines 9-13 & 35-41; column 18, lines 54-65);"

Applicant argues that DaCosta teaches a limited method of notifying a user if a script fails and disabling the entire script (col. 6, lines36-38 and col. 18, lines 63-67). Clearly, DaCosta fails to teach indicating a point of failure in the process or creating a data file containing parameters of the failure as espoused by the Examiner. Nowhere in DaCosta is a teaching of singling out where the error occurs in a navigating script when sending an error notification, creating a data file or including parameters of the failure in the data file, as claimed.

The Examiner admits that DaCosta fails to teach storing the data file in a data repository with a point-of-failure indication in the routine and an identifier of the associated Web page. The Examiner relies upon Weinberg to teach said limitation. Applicant points out that Weinberg teaches testing and verifying outputs of a server and has no motivation for testing a navigation routine, marking a point of failure in the routine, as claimed. Weinberg teaches steps for submitting and receiving data from a transactional server and marks the step that fails; i.e. no data return or unexpected data return (col. 15, lines 27-34; 47-55; Fig. 5F; col. 16, lines 7-14). Therefore, applicant has clearly shown that the art of Weinberg is merely marking steps when a transaction server returns data that is not expected. Weinberg is not capable of marking a point of failure in a navigation routine, as claimed.

Further, applicant points out that part of the subject matter relied upon in DaCosta (US 6,826,553) is not valid prior art. The Examiner admits that the DaCosta reference alone would not provide an appropriate date required for proper prior art. The Examiner states; "However in view of the claimed priority to the continuation-in-part application (i.e. 09/465,028) and the provisional applications (i.e. 60/147,875 & 60/112,769), the cited disclosure is given the appropriate priority date that it deserves. The Examiner also recognizes that the burden now lies with the Applicant who must point out, outside of mere allegations, what limitations are missing from the priority art references that are relied upon in the present application. While the Examiner will not specifically point out each relied upon citation, the Examiner respectfully believes that the cited subject matter from the DaCosta reference should be given the date of the provisional application 60/112,769 or at the least that of the CIP application. In general, the key limitations relied upon in the DaCosta reference (i.e. information pertaining to Figs. 1 and 7) are clearly exhibited in the priority references."

Applicant argues that the Examiner should at least provide a valid reference and point out in that reference the teachings that read on applicant's claim limitations. It is not applicant's burden to do so. The Examiner at least relies upon column 18 lines 26-32 to read on dependent claims 4, and 14-17. Applicant points out that the reference

09/465,028, which the Examiner relies upon for a priority date only discloses Figures and teaching up to Figure 16. Column 18, lines 26-32 of DaCosta is new matter and therefore cannot be relied upon by the Examiner to reject applicant's claimed subject matter.

Regarding the provisional applications, it is applicant's understanding that these documents are only afforded their publishing date, not the filing date, as provisional applications are not in the public domain until they are published. Applicant believes the Examiner has the responsibility to provide the prior art, stating the valid priority date and point out in the reference where applicant's claim limitations are taught or allow the claim.

Based on the above arguments regarding Dacosta, Weinberg and applicant's right to have valid prior art, applicant believes claims 1, 12 and 18 are patentable over the art of DaCosta and Weinberg. Dependent claims 2-11, 13-17 and 19-28 are patentable on their own merits, or at least as depended from a patentable claim.

If there are any time extensions needed beyond any extension specifically requested with this amendment, such extension of time is hereby requested. If there are any fees due beyond any fees paid with this amendment, authorization is given to deduct such fees from deposit account 50-0534.

Respectfully Submitted,
Tim Armandpour et al.

By /Donald R. Boys/
Donald R. Boys
Reg. No. 35,074

Central Coast Patent Agency, Inc.
3 Hangar Way, Suite D
Watsonville, CA 95076
831-768-1755